

# APLIKASI WEB SEMANTIK UNTUK PENCARIAN MATERI PERKULIAHAN

<sup>1</sup>Ferdila (10107686),

<sup>2</sup>Metty Mustikasari, SKom., MSC.

<sup>1</sup>Mahasiswa Sistem Informasi Universitas Gunadarma

Ferdi\_686@yahoo.co.id

<sup>2</sup>Dosen Tetap Universitas Gunadarma

metty@staff.gunadarma.ac.id

## ABSTRAK

Penulisan ini membahas tentang pembuatan aplikasi pencarian materi perkuliahan untuk mahasiswa tingkat satu jurusan sistem informasi universitas Gunadarma. Aplikasi ini dibuat menggunakan teknologi web semantik. Dalam penulisan ini komponen-komponen teknologi web semantik yang dipergunakan antara lain RDF (Resource Description Framework) dipergunakan sebagai representasi pengetahuan yang digunakan. Komponen yang kedua adalah SPARQL yang dipergunakan sebagai query untuk mengambil informasi yang terdapat pada RDF. Komponen yang terakhir adalah RAP (RDF API for PHP) yang dipergunakan untuk menjembatani antara RDF dengan PHP sehingga informasi yang ada di dalam RDF dapat digunakan pada PHP. Aplikasi ini dirancang untuk menghasilkan keluaran yang mudah dimengerti oleh pemakai serta membantu dalam pencarian materi perkuliahan. Setelah aplikasi tersebut di uji coba, menunjukkan bahwa aplikasi ini menghasilkan keluaran yang mudah dimengerti oleh pengguna namun pada kemudahan penggunaan masih dirasa kurang. Dikarenakan masih kurangnya fasilitas untuk mempermudah pencarian materi perkuliahan.

Kata kunci: Ontologi, Pencarian materi perkuliahan , RAP , RDF ,web semantik

## ABSTRACT

This writing is about making college search application materials for freshmen majoring in information systems at the University Gunadarma. This application is created using semantic web technology. In this paper the components of semantic web technologies are used such as RDF (Resource Description Framework) is used as an ontology that represents knowledge used. The second component is used as a SPARQL query to retrieve information contained in the RDF. The last component is the RAP (RDF API for PHP) is used to bridge between RDF with PHP so that the information in the RDF can be used on PHP. This application is designed to produce output that is easily understood by the user and assist in the search for lecture material. Once the application is tested, showing that this application produce output that is easily understood by

users, but on ease of use is deemed lacking. Due to still the lack of facilities to facilitate the search for lecture material.

Keywords: Ontology, lecture note retrieval, RAP, RDF, Semantic Web

## PENDAHULUAN

Internet merupakan salah satu sumber informasi yang banyak dipakai di jaman sekarang ini. Terdapat berbagai macam informasi yang dapat ditemui dan tersimpan di internet. Namun informasi tersebut terkadang sulit untuk dicari sesuai dengan kebutuhan walaupun tersedia. Penyaringan informasi biasanya dilakukan oleh pemakai dengan cara memilah-milah informasi apa saja yang ia butuhkan. Web semantik adalah informasi dalam jumlah sangat besar di *World Wide Web* yang terhubung secara global dengan suatu cara tertentu dan dimengerti atau dipahami oleh mesin, sehingga dapat diproses secara langsung oleh mesin menjadi *knowledge* untuk ditampilkan kepada pengguna[1]. Sehingga hal tersebut dapat mempermudah pengguna dalam mendapatkan informasi sesuai yang diinginkannya dengan cara yang mudah.

Pencarian materi perkuliahan di Gunadarma dapat dilakukan dengan menggunakan staffsite gundarma. Pada halaman staffsite dapat melakukan pencarian materi perkuliahan sesuai dengan nama dosen yang mengajar suatu mata kuliah. Namun belum tersedia layanan pencarian dengan menggunakan kata kunci berdasarkan mata kuliah ataupun materi perkuliahan. Oleh karena itu penulis mencoba untuk membuat sebuah aplikasi berbasis web menggunakan teknologi semantik yang dipergunakan untuk mencari materi perkuliahan.

## TINJAUAN PUSTAKA

### Web Semantik

Secara garis besar web semantik adalah informasi dalam jumlah sangat besar di *World Wide Web* yang terhubung secara global dengan suatu cara tertentu dan dimengerti atau dipahami oleh mesin, sehingga dapat diproses secara langsung oleh mesin menjadi *knowledge* untuk ditampilkan kepada pemakai. Web semantik juga dapat dikatakan sebagai sebuah cara yang efisien untuk merepresentasikan data di *World Wide Web* sebagai sebuah *database* yang terhubung secara global. Istilah web semantik pertama kali dikemukakan oleh Tim Berners-Lee yang merupakan penemu *World Wide Web*[1].

Dengan metode web semantik, data berbasis HTML dapat dirubah menjadi format yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan antara informasi. Web semantik mampu melakukan perubahan ini dengan bantuan XML (Extensible Markup Language) dan *data language standards* seperti RDF (Resource Description Framework) dan OWL (Ontology Web Language), dua standarisasi dari W3C (World Wide Web Consortium). Dengan berbagai standar tersebut, memungkinkan pengembang web (Web Developer) untuk menambahkan satu layer "arti" pada dokumen webnya. Sebagai *framework*

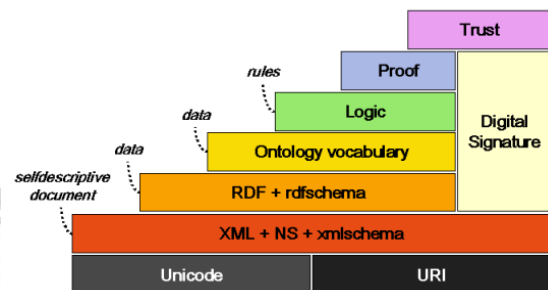
untuk mendefinisikan bagaimana beberapa data terhubung dan bagaimana relasi yang menyertai data-data tersebut seharusnya ditampilkan[1].

Web semantik bukanlah *Artifitial Intelegant* (kecerdasan buatan), karena mesin tidak dengan sendirinya memahami bahasa manusia secara menyeluruh. Konsep ini hanya menandakan kemampuan mesin untuk memecahkan *well-defined problems* (permasalahan yang telah ditentukan) dengan cara melakukan *well-defined operations* (operasi untuk memecahkan masalah yang juga telah ditentukan) pada *well-defined data* (data yang juga telah ditentukan) yang tersedia. Jadi, untuk bahasa manusia yang berada di luar *well-defined data*, mesin sudah tidak mampu lagi untuk memahami bahasa tersebut[2].

## Arsitektur Web Semantik

Prinsip-prinsip Web semantik diimplementasikan dalam lapisan teknologi Web dan standar. Lapisan disajikan pada Gambar 2.1 . Lapisan Unicode dan URI memastikan bahwa kita menggunakan set karakter internasional dan menyediakan sarana untuk mengidentifikasi objek dalam Web semantik. Lapisan XML dengan definisi skema *namespace* dan pastikan kita dapat mengintegrasikan definisi Web semantik dengan standar berbasis XML lainnya. Dengan RDF [RDF] dan RDFSchema [RDFS] adalah mungkin untuk membuat pernyataan tentang objek dengan URI dan menentukan itu kosakata yang dapat disebut dengan URI. Ini adalah lapisan di mana kami dapat memberikan jenis sumber daya dan link. Lapisan Ontologi mendukung evolusi kosakata karena dapat menentukan hubungan antara konsep-

konsep yang berbeda. Dengan lapisan *Digital Signature* untuk mendeteksi perubahan pada dokumen[3].



Gambar 1 Arsitektur Web Semantik [3]

Lapisan atas: Logika, Bukti dan *Trust*, saat ini sedang diteliti dan demonstrasi aplikasi sederhana sedang dibangun. Lapisan Logika memungkinkan penulisan aturan sedangkan lapisan Bukti mengeksekusi aturan dan mengevaluasi bersama-sama dengan mekanisme lapisan *Trust* untuk aplikasi apakah untuk mempercayai bukti yang diberikan atau tidak[3].

## Ontologi

Istilah ontologi sebenarnya berasal dari istilah filosofi “ontologi” yang artinya sesuatu yang sesungguhnya ada dan bagaimana menggambarannya. Dalam dunia komputer ontologi digunakan untuk menspesifikasikan suatu konseptualisasi. Dalam istilah lain ontologi dijelaskan sebagai suatu representasi dari domain pengetahuan tertentu yang berisi istilah-istilah dalam domain tersebut beserta hubungan antara istilah-istilah yang ada[4].

Ontologi saat ini banyak digunakan terutama untuk mendukung web semantik, yaitu teknologi web yang diarahkan dapat memahami makna suatu kata atau kalimat yang diberikan oleh pengguna. Membuat komputer mengerti seperti manusia adalah

suatu hal yang sepertinya mustahil, namun visi ini terus diupayakan dengan menyediakan seperangkat alat sehingga membuat mesin atau komputer dengan mudah dapat memproses informasi dan mengerti informasi yang diinginkan oleh pengguna[4].

Tidak ada standar khusus untuk membangun suatu ontologi dan tidak ada justifikasi bahwa ontologi yang dikembangkan oleh seseorang adalah salah atau benar. Kualitas ontologi dapat dilihat dari aplikasi yang dibangun berdasarkan ontologi ini. Ketika aplikasi yang dibangun dapat memenuhi kebutuhan pengguna dan menjawab permasalahan yang ada maka ontologi yang digunakan termasuk ontologi yang berkualitas[4].

### Bahasa Ontologi

Ontologi sendiri mempunyai struktur bahasa yang formal (terdefinisi), agar dapat digunakan. Beberapa struktur bahasa yang menyusun ontologi antara lain [5] XML (Extensible Markup Language) merupakan Struktur mirip HTML yang *tag*-nya dapat didefinisikan sendiri. XML Schema, sebagai bahasa yang membatasi struktur yang didefinisikan pada dokumen XML. RDF (Resource Description Framework) digunakan sebagai model data untuk objek ('resources') dan relasi diantaranya, menyediakan semantik yang sederhana untuk model data tersebut, dan data model ini dapat disajikan dalam sintaksis XML. RDF Schema Adalah kosakata untuk menjelaskan *properties* dan *classes* dari sumber RDF, dengan sebuah *semantics* untuk hirarki penyamarataan dari *properties* dan *classes*. Komponen terakhir adalah OWL (Ontology Web Language) Menambahkan beberapa kosakata untuk

menjelaskan *properties* dan *classes*, antara lain : relasi antara *classes* (misalkan disjointness), kardinalitas (misalkan 'tepat satu'), *equality*, berbagai tipe dari *properties*, karakteristik dari *properties* (misalkan *symmetry*), menyebutkan satu persatu *classes*.

### RDF (Resource Description Framework)

Dalam penelitian ini menggunakan RDF sebagai bahasa ontologi untuk merepresentasikan pengetahuan yang ada sebagai sumber informasi. RDF merupakan suatu metadata yang digunakan untuk mendeskripsikan alamat sumber daya pada web[6]. RDF memiliki bagian-bagian standar yang digunakan untuk membentuknya antara lain *resource* yang digunakan untuk menggambarkan apa saja yang dapat dimiliki oleh sebuah URI, *Property* berisikan nama *property-property* yang ada dalam *resource*, *property value* yang berisikan nilai dari sebuah *property*. Dari bagian-bagian tersebut nantinya dapat membentuk sebuah *statement* yang berisikan *resource*, *property* dan *property value* atau yang di kenal sebagai *triple* (3-tuple) yang berfungsi sebagai penyimpanan data dan relasi antar data[7]. Berikut merupakan contoh bentuk RDF[8].

```
<?xml version="1.0"?>
<RDF>
  <Description
    about="http://www.w3schools.com/rdf">
    <author>Jan Egil Refsnes</author>

    <homepage>http://www.w3schools.com</h
omepage>
  </Description>
</RDF>
```

Pada RDF diatas terdapat beberapa komponen-komponen yang menyusunnya



antara lain. Resource yang digunakan pada RDF tersebut adalah <http://www.w3schools.com/rdf> yang merupakan URI. Terdapat juga property yaitu *author* dan juga *homepage*, pada property tersebut juga mempunyai nilai yang disebut dengan property value. Property value untuk *author* adalah Jan Egil Refsnes dan property value untuk *homepage* adalah <http://www.w3schools.com>. Dari komponen-komponen tersebut dapat dibentuk menjadi statemen antara lain[27].

Statement 1: <http://www.w3schools.com/rdf> memiliki *author* yaitu Jan Egil Refsnes

Pada statement ini terdapat 3 komponen yaitu subjek, dimana yang menjadi subjek adalah resource yang ada pada RDF yaitu <http://www.w3schools.com/rdf>. Sedangkan yang berfungsi sebagai predikat adalah *property* yang terdapat pada RDF yaitu *author*. Serta objek adalah *property value* yaitu Jan Egil Refsnes[8].

Statement 2: <http://www.w3schools.com/rdf> memiliki *homepage* yaitu <http://www.w3schools.com>

Pada statement tersebut memiliki komponen-komponen yang sama yaitu subjek predikat dan objek sesuai dengan *statement* tersebut[8].

## RAP (RDF API for PHP)

RAP merupakan tools web semantik untuk bahasa PHP yang memiliki fitur untuk memanipulasi, menyimpan, melakukan *query* serta membangun graph RDF[9]. RAP diawali sebagai proyek *open source* oleh Freie Universitat Berlin pada 2002 dan telah mengalami perkembangan di komunitas web semantik. Inti dari RAP meliputi dua implementasi atas penyimpanan *statement*,

yaitu apakah menyimpan *graph* RDF dalam *memory* sistem maupun dalam *database relational*[10].

## SPARQL

Model data RDF berupa suatu statemen dalam bentuk triple yang terdiri dari subjek, predikat, dan objek. Untuk mendapatkan informasi dari suatu graph RDF dibutuhkan suatu *query*. SPARQL merupakan suatu bahasa *query* yang dapat digunakan untuk mengakses data pada web semantic. Dengan menggunakan SPARQL memungkinkan untuk melakukan beberapa hal antara lain. Mengambil nilai dari data yang terstruktur maupun data yang semi terstruktur. Mengembangkan data dengan melakukan *query* terhadap suatu relasi yang tidak diketahui. Dapat melakukan *query* operasi *join* yang kompleks pada *database* yang berlainan secara lebih sederhana. Mengubah suatu data RDF menjadi vocabulary yang lain. Hasil dari *query* SPARQL dapat mengembalikan nilai dalam beberapa format data yang antara lain : XML, RDF, dan HTML [10].

Berikut ini adalah contoh sederhana dari SPARQL :

Data :

```
<http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title>
"SPARQL Tutorial"
```

Query :

```
SELECT ?title
WHERE
<http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title>
?title .
```

Tabel 1. Hasil Query

[title]
"SPARQL Tutorial"

Variabel SPARQL dimulai dengan tanda "?" dan merupakan suatu *node (resource* atau *literal)* di dalam RDF *triple*. Sedangkan pernyataan "SELECT" mengembalikan suatu tabel dari variabel dan nilai yang dideskripsikan di dalam *query*[10].

## METODE PENELITIAN

Pada penelitian ini menggunakan beberapa tahapan dan metode dalam penyusunannya. Metode yang pertama adalah studi pustaka. Pada bagian ini dilakukan pencarian yang berkaitan dengan penelitian ini antara lain semantik web, ontologi, RDF (Resource Description Framework), RAP (RDF API for PHP) serta SPARQL.

Setelah melakukan studi pustaka terhadap teori-teori yang sudah dijelaskan diatas, langkah yang kedua adalah perancangan RDF. Kemudian diperlukan query yang berfungsi untuk mengambil pengetahuan yang ada pada RDF tersebut dengan menggunakan SPARQL. Setelah RDF dan SPARQL dibentuk maka diperlukan interface yang akan disajikan kepada pengguna pada aplikasi ini di bagi menjadi beberapa bagian interface antara lain tampilan halaman depan, halaman pencarian detail, halaman hasil pencarian, halaman hasil pencarian detail, halaman materi serta halaman kontak dan *about*.

Aplikasi yang sudah terbentuk dari tahapan diatas kemudian dilakukan uji coba. Uji coba terdiri dari dua bagian yaitu uji coba dari sisi pengembang serta uji coba dari sisi pengguna untuk melihat apakah aplikasi ini sudah bias berjalan sesuai dengan fungsinya.

## PEMBAHASAN

### Perancangan Ontologi

Pada aplikasi pencarian materi perkuliahan ini menggunakan teknologi web semantik sehingga diperlukan sebuah ontologi yang berguna merepresentasikan pengetahuan apa saja yang diperlukan untuk membangun aplikasi pencarian materi perkuliahan tersebut.

Ontologi tersebut berisi kriteria yang telah diklasifikasikan, dimana klasifikasi tersebut berupa hal abstrak yang memiliki nilai di dunia nyata. Pada struktur ontologi MateriPerkuliahan yang digunakan pada aplikasi mempunyai standar stuktur sebagai berikut:

1. Judul, digunakan untuk menyimpan nama judul materi dari sebuah MateriPerkuliahan
2. MataKuliah, digunakan untuk menyimpan nama mata kuliah dari sebuah MateriPerkuliahan
3. Dosen, digunakan untuk menyimpan nama dosen yang mengajar dan memiliki materi perkuliahan tersebut pada *staffsite*
4. LinkStaff, digunakan untuk menyimpan alamat URL dari materi perkuliahan yang dimiliki dosen pada *staffsite*-nya
5. LinkView, digunakan untuk menyimpan alamat URL untuk menampilkan materi yang dimiliki dosen.

Struktur diatas nantinya diubah menjadi bentuk RDF yang merupakan salah satu struktur bahasa ontologi. Sehingga dapat dipergunakan dalam web semantik dan menampilkan informasi yang dibutuhkan dalam pencarian materi perkuliahan.

Untuk merubah struktur tersebut kedalam model RDF diperlukan beberapa bagian penting. Bagian-bagian penting dalam model RDF yaitu *Resource* yang digunakan untuk menggambarkan apa saja yang dapat dimiliki oleh sebuah URI, *Property* berisikan nama *property-property* yang ada dalam *resource*, *property value* yang berisikan nilai dari sebuah *property*. Dari bagian-bagian tersebut nantinya dapat membentuk sebuah statemen yang berisikan *resource*, *property* dan *property value* atau yang di kenal sebagai *triple (3-tuple)*.

Sesuai dengan standar model RDF, Struktur diatas dapat diubah menjadi bentuk RDF dengan bentuk sebagai berikut:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/
22-rdf-syntax-ns#"
  xmlns:ns1="http://www.w3.org/2001/vc
ard-rdf/3.0#"
  xmlns:owl="http://www.w3.org/2002/07
/owl#">
  <rdf:Description
    rdf:about="http://ferdi.com/ontologies/c
oba.owl#AP1A_MGG_1">
    <ns1:isA>Materi</ns1:isA>
    <ns1:hasJudul>Pendahuluan</ns1:hasJu
dul>
    <ns1:hasMataKuliah>Alogaritma
Pemograman 1A</ns1:hasMataKuliah>
    <ns1:hasDosen>Marliza      Ganefi
G</ns1:hasDosen>
    <ns1:hasLinkStaff>http://marliza.staff.g
unadarma.ac.id/Downloads/files/9901/Pe
ngenalan+Algoritma.ppt</ns1:hasLinkSt
aff>
```

```
<ns1:hasLinkView>http://docs.google.c
om/viewer?url=http%3A%2F%2Fmarliz
a.staff.gunadarma.ac.id%2FDownloads
%2Ffiles%2F9901%2FPengenalan%2B
Algoritma.ppt</ns1:hasLinkView>
</rdf:Description>
</rdf:RDF>
```

Pada bentuk RDF diatas digambarkan bahwa model RDF tersebut memiliki resource dari URI [http://ferdi.com/ontologies/coba.owl#AP1A\\_MGG\\_1](http://ferdi.com/ontologies/coba.owl#AP1A_MGG_1). Di

dalam *resource* tersebut terdapat *property-property* seperti *isA*, *hasJudul*, *hasMataKuliah*, *hasDosen*, *HasLinkStaff*, *hasLinkView* yang mempunyai *prefix* yang berasal dari *namespace* *ns1*. Property tersebut juga memiliki *property value* sebagai contoh *property* *hasJudul* memiliki *property value* *pendahuluan*.

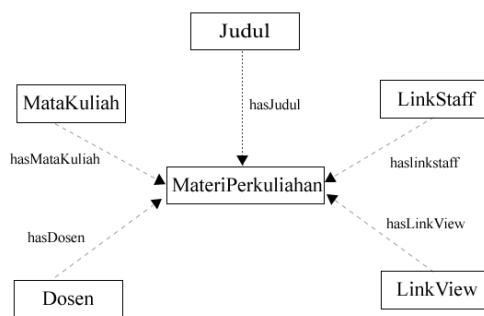
Untuk membentuknya kedalam sebuah statemen RDF (*triple*) maka diperlukan bagian-bagian yang diterjemahkan sebagai subjek, predikat dan objek. Dalam hal ini subjek pada RDF tersebut adalah *resource* yang terdapat dalam RDF. Sedangkan Objek pada RDF tersebut adalah *property* dari RDF dan objeknya merupakan *property value* RDF. Berikut merupakan contoh statemen dari RDF diatas.

[http://ferdi.com/ontologies/coba.owl#AP1A\\_MGG\\_1](http://ferdi.com/ontologies/coba.owl#AP1A_MGG_1) *hasJudul* *pendahuluan*

P O S

Setiap *resource* dari RDF tersebut memiliki enam *property* sehingga dapat dibentuk enam statemen dimana *resource* tersebut menjadi subjek yang merupakan dan predikat

diambil dari *property* yang tersedia serta objek diambil dari *property value* yang ada. Dengan terbentuknya statemen maka penggambaran visual dari sebuah RDF atau yang disebut dengan *graph* dapat terlihat. Berikut penggambaran struktur graph dari RDF MateriPerkuliahan.



Gambar 2. Struktur RDF Graph MateriPerkuliahan

## Perancangan Query SPARQL

Model data RDF yang telah terbentuk terdiri dari statemen yang di dalamnya terdiri dari subjek, predikat dan objek. Untuk mendapatkan informasi dari sebuah RDF dibutuhkan suatu *query*. SPARQL merupakan suatu bahasa *query* yang dapat digunakan untuk mengakses informasi yang ada pada suatu RDF.

Query SPARQL terdiri dari beberapa bagian utama yaitu bagian *select*, *from* dan *where*. Pada bagian *select* berisikan variabel yang digunakan untuk mengambil nilai yang diinginkan. Penulisan variabel pada bagian *select* diawali dengan simbol tanda Tanya (?). bagian yang kedua adalah bagian *from*, bagian ini berfungsi menunjuk kebagian *resource* yang ingin diambil informasinya. Sedangkan pada bagian *where* dipergunakan sebagai menunjuk kebagian *sub graph* yang ingin diambil informasinya.

Pada aplikasi ini memerlukan *query* yang mengambil informasi dari ontologi yang tersedia. Informasi yang diperlukan antara lain judul materi, nama dosen, nama mata kuliah, link download materi serta link untuk melihat materi dari sebuah materi. *Query* yang dipergunakan mempunyai struktur seperti yang dijelaskan diatas namun pada bagian *from* tidak dipergunakan karena *resource* yang dipergunakan hanya satu dalam RDF tersebut. Berikut merupakan penjelasan *query-query* apa saja yang diperlukan dalam aplikasi pencarian materi perkuliahan beserta bentuk *query* SPARQL yang digunakan.

1. *Query* untuk mendapatkan informasi mengenai materi perkuliahan berupa judul materi, mata kuliah, dosen yang mengajar dan memiliki materi tersebut, alamat URL materi dosen tersebut, serta alamat URL untuk menampilkan materi.

Bentuk query SPARQL:

```

select ?judul ?matakuliah ?dosen
?linkstaff ?linkview
WHERE {
    ?x vcard:hasJudul ?judul .
    ?x vcard:hasMataKuliah
?matakuliah .
    ?x vcard:hasDosen ?dosen .
    ?x vcard:hasLinkStaff
?linkstaff .
    ?x vcard:hasLinkView
?linkview .
    FILTER regex (?judul,
"Nama_Judul_Materi_Yang_Dic
ari", "i")
}

```

2. *Query* untuk menampilkan informasi mengenai materi perkuliahan yang dipergunakan pada halaman materi serta digunakan sebagai acuan untuk menampilkan materi. Bentuk query SPARQL:

```

select *
WHERE {

```



```

?x vcard:hasJudul ?judul .
?x      vcard:hasMataKuliah
?matakuliah .
?x vcard:hasDosen ?dosen .
?x      vcard:hasLinkStaff
?linkstaff .
?x      vcard:hasLinkView
?linkview .
FILTER      (?judul      =
"Nama_Judul_Materi_Yang_Dicari"
&&      ?matakuliah      =
"Nama_Matakuliah_Yang_Dicari"
&&      ?dosen      =
>Nama_Matakuliah_Yang_Dicari
")})

```

3. *Query* yang digunakan untuk menampilkan informasi mengenai materi perkuliahan pada halaman Pencarian Detail. Pada halaman Pencarian Detail terdapat beberapa kata kunci yang dapat dimasukan oleh pemakai sehingga mempengaruhi bentuk *query* SPARQL yang dipergunakan. *Query* yang dipergunakan berubah sesuai dengan kata kunci yang diinput oleh pemakai, namun perubahan *query* tersebut hanya terdapat pada bagian FILTER. Pada bagian FILTER terlebih dahulu disimpan variabel yang berisi penggalan statemen FILTER sesuai dengan input kata kunci yang digunakan. Bentuk *query* SPARQL jika kata kunci yang dimasukan judul materi dan mata kuliah.

```

select *
WHERE {
?x vcard:hasJudul ?judul .
?x      vcard:hasMataKuliah
?matakuliah .
?x vcard:hasDosen ?dosen .
?x      vcard:hasLinkStaff
?linkstaff .

```

```

?x      vcard:hasLinkView
?linkview .

```

```

FILTER      regex      (?judul,
"Nama_Judul_Materi_Yang_Dicari", "i")
&&      regex      (?matakuliah,
"Nama_Matakuliah_Yang_Dicari", "i")})

```

### Perancangan Tampilan Aplikasi

Rancangan tampilan dipergunakan untuk memperkirakan tampilan aplikasi yang diinginkan. Pada aplikasi ini dibutuhkan beberapa halaman untuk menampilkan informasi yang dibutuhkan oleh pengguna sehingga terdapat beberapa rancangan tampilan aplikasi yang dipergunakan. Sebelum rancangan tampilan aplikasi dibuat terlebih dahulu ditetapkan halaman apa saja yang akan dipergunakan dalam aplikasi ini untuk menunjang penyajian informasi yang ingin dihasilkan. Pada aplikasi ini ditetapkan menggunakan 7 halaman *web page* antara lain. Rancangan tampilan halaman depa adalah tampilan halaman yang pertama kali muncul pada saat aplikasi dijalankan. Pada halaman tersebut terdapat menu serta fasilitas pencarian materi perkuliahan menggunakan kata kunci berupa nama matakuliah, nama dosen yang mengajar atau judul materi perkuliahan yang ingin dicari.

Banner			
Menu Halaman Depan	Menu Pencarian Detail	Menu Kontak	Menu Tentang
Label kata kunci	input text kata kunci	tombol kata kunci	
Informasi singkat tentang aplikasi			
footer			

Gambar 3. Rancangan Halaman Depan

Rancangan tampilan yang kedua adalah rancangan halaman hasil pencarian. Halaman ini akan muncul jika pencarian dilakukan pada halaman depan sehingga menghasilkan output berupa judul materi perkuliahan, judul mata kuliah, nama dosen yang mengajar mata kuliah tersebut, link download serta link view.

The wireframe shows a page layout with a 'Banner' at the top. Below it is a navigation bar with four links: 'Menu Halaman Depan', 'Menu Pencarian Detail', 'Menu Kontak', and 'Menu Tentang'. The main content area contains a search bar with a label 'Label kata kunci', an input field 'input text kata kunci', and a button 'tombol kata kunci'. Below the search bar is a large box labeled 'Hasil pencarian'. At the bottom of the main content area is a link labeled 'link halaman hasil pencarian'. The footer section contains the word 'footer'.

Gambar 4. Rancangan Halaman Hasil Pencarian

Rancangan tampilan selanjutnya adalah rancangan pencarian detail, pada halaman ini berfungsi untuk mencari materi perkuliahan menggunakan lebih dari satu kata kunci. Pada halaman ini kata kunci dapat berupa nama dosen, judul materi, dan mata kuliah. Selain itu pada pencarian ini dapat menggunakan operator and dan or dalam melakukan pencarian dengan beberapa kata kunci yang disediakan.

The wireframe shows a page layout with a 'Banner' at the top. Below it is a navigation bar with four links: 'Menu Halaman Depan', 'Menu Pencarian Detail', 'Menu Kontak', and 'Menu Tentang'. The main content area contains a search bar with four labels 'Label kata kunci 1' through 'Label kata kunci 4', four corresponding input fields 'input text kata kunci 1' through 'input text kata kunci 4', and a button 'tombol kata kunci'. Below the search bar is a large box labeled 'Hasil pencarian'. At the bottom of the main content area is a link labeled 'link halaman hasil pencarian'. The footer section contains the word 'footer'.

Gambar 5. Rancangan Halaman Pencarian Detail

Setelah pencarian dilakukan pada halaman pencarian detail maka hasil pencarian tersebut akan ditampilkan pada halaman hasil pencarian detail. Halaman tersebut memuat hasil pencarian sama halnya dengan halaman hasil pencarian, namun pada halaman tersebut dapat dilakukan pencarian kembali dengan menggunakan beberapa kata kunci. Hasil pencarian yang dilakukan mempunyai link yang menuju pada halaman materi pada halaman ini berisi tampilan materi serta info mengenai materi tersebut.

The wireframe shows a page layout with a 'Banner' at the top. Below it is a navigation bar with four links: 'Menu Halaman Depan', 'Menu Pencarian Detail', 'Menu Kontak', and 'Menu Tentang'. The main content area contains a search bar with four labels 'Label kata kunci 1' through 'Label kata kunci 4', four corresponding input fields 'input text kata kunci 1' through 'input text kata kunci 4', and a button 'tombol kata kunci'. Below the search bar is a large box labeled 'Hasil pencarian'. At the bottom of the main content area is a link labeled 'link halaman hasil pencarian'. The footer section contains the word 'footer'.

Gambar 6. Rancangan Halaman Hasil Pencarian Detail

Setelah hasil pencarian diperlihatkan baik pada pencarian dengan menggunakan satu kata kunci atau dengan menggunakan pencarian detail. Hasil dari pencarian tersebut beberapa memiliki link yang berfungsi untuk melihat materi yang tersedia. Namun halaman ini hanya dapat diakses pada file-file yang memiliki format .pdf, .doc ataupun .ppt, sedangkan untuk file yang memiliki file extensi .zip atau .rar halaman ini tidak tersedia.

Gambar 7. Rancangan Halaman Materi

Selain itu terdapat juga dua halaman lain selain halaman yang sudah dijabarkan sebelumnya yaitu halaman kontak yang berfungsi sebagai pengirim pesan kepada pembuat aplikasi. Pada halaman ini terdapat beberapa field yang harus diisi antara lain field nama, subjek, email serta pesan yang berfungsi sebagai media penyampai pesan serta komunikasi dengan pembuat aplikasi.

Gambar 8. Rancangan Halaman Kontak

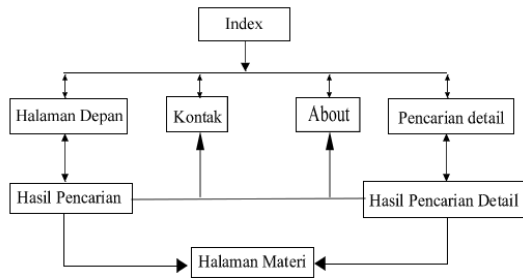
Rancangan halaman yang terakhir adalah halaman *about* yang berisi informasi mengenai aplikasi ini. Pada halaman ini terdapat informasi tentang tujuan dibuatnya aplikasi ini serta teknologi yang digunakan dalam aplikasi ini. Selain itu terdapat biodata pembuat aplikasi ini.

Gambar 9. Halaman *About*

### Struktur navigasi

Struktur navigasi merupakan urutan mengakses halaman satu ke halaman selanjutnya. Struktur navigasi digunakan untuk menggambarkan fungsi-fungsi yang ada pada seluruh halaman sistem yang dibuat. Struktur navigasi yang digunakan

adalah struktur navigasi campuran (hierarki). Berikut adalah penggambarannya:



Gambar 10. Struktur Navigasi

## Implementasi

Teknologi pendukung yang digunakan untuk melakukan implementasi Teknologi tersebut terdiri dari perangkat lunak serta perangkat keras yang digunakan ketika implementasi aplikasi tersebut. Berikut merupakan spesifikasi perangkat lunak dan perangkat keras ketika aplikasi ini diimplementasikan.

1. RAP Library versi 9.6
2. Apache versi 2.2.14 sebagai web server
3. PHP 5.3.1
4. Windows XP.
5. Mozilla Firefox sebagai web browser
6. Prosesor Intel Pentium Dual Core 2.0 GHz
7. RAM DDR 1024 MB
8. Hardisk 80 GB

## Uji Coba

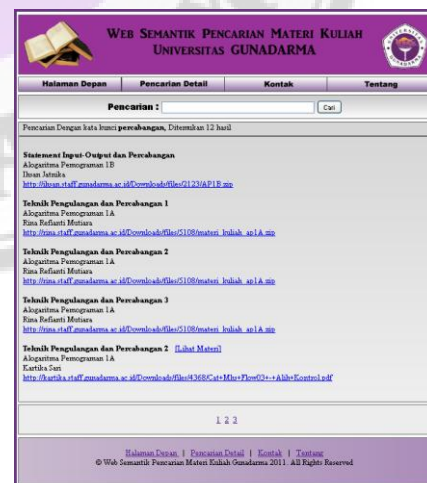
Uji coba dilakukan oleh Pengembang dan Pengguna, Pengembang melakukan uji coba

untuk mengetahui apakah aplikasi yang telah dikembangkan sudah berjalan sesuai dengan fungsi yang diinginkan dan apakah aplikasi sudah siap untuk diujicobakan secara terbatas atau belum, sedangkan uji coba oleh Pengguna bertujuan untuk mengetahui apakah tujuan Pengembangan aplikasi sudah tercapai atau belum.

## Uji Coba Pengembangan

Uji coba dilakukan dengan cara melakukan pencarian materi perkuliahan dengan menggunakan aplikasi ini. Pengujian dilakukan pada bagian pencarian serta pencarian detail. Pada bagian pencarian detail pengujian dilakukan dengan cara memasukkan kata kunci lebih dari satu serta menguji pencarian menggunakan operator yang berbeda yaitu operator and dan or.

Uji coba yang pertama dilakukan dengan cara melakukan pencarian pada halaman depan dimana pada halaman tersebut kata kunci yang dimasukan hanya dapat berupa salah satu dari nama materi, nama mata kuliah atau nama dosen. Pada pengujian pencarian ini digunakan kata kunci yaitu “percabangan”. Berikut tampilan ketika pencarian dilakukan.



Gambar 11. Hasil Pencarian



Pada gambar diatas memperlihatkan hasil dari pencarian dengan kata kunci “percabangan” dimana dengan menggunakan kata kunci tersebut mendapatkan 12 hasil pencarian.

Selanjutnya uji coba dilakukan pada pencarian detail. Pada pencarian detail uji coba dilakukan dengan menggunakan beberapa kata kunci dan juga menggunakan kedua operator yang tersedia dan menguji pencarian pada ketiga kata kunci tersebut. Pada pengujian yang pertama pada pencarian detail dilakukan dengan cara memasukan kata kunci judul materi “percabangan” dan nama dosen “Ihsan Jatnika” serta menggunakan operator or. Berikut tampilan ketika pengujian pertama pada pencarian detail.

Gambar 12. Hasil Pencarian Detail Operator OR

Pada gambar diatas memperlihatkan hasil dari pencarian dengan kata kunci judul materi “percabangan” dan nama dosen “Ihsan Jatnika” dan menggunakan operator OR mendapatkan 33 hasil pencarian.

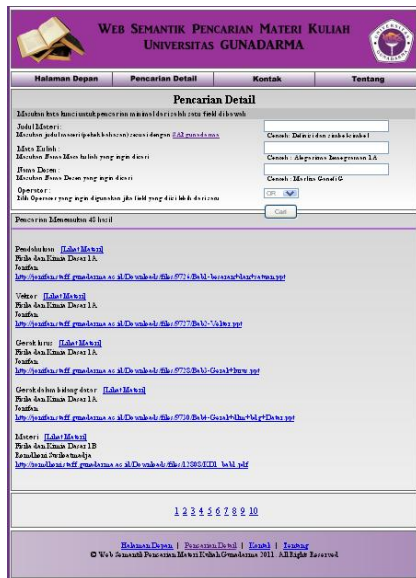
Pengujian kedua pada pencarian detail juga dilakukan uji coba dengan cara memasukan

kata kunci kunci judul materi “percabangan” dan nama dosen “Ihsan Jatnika” dan menggunakan operator AND. Berikut tampilan ketika pengujian pada pencarian detail dengan menggunakan 2 kata kunci serta operator AND.

Gambar 13. Hasil Pencarian Detail Operator AND

Pada gambar diatas memperlihatkan hasil dari pencarian dengan kata kunci judul materi “percabangan” dan nama dosen “Ihsan Jatnika” dan menggunakan operator AND hanya mendapatkan 1 hasil pencarian.

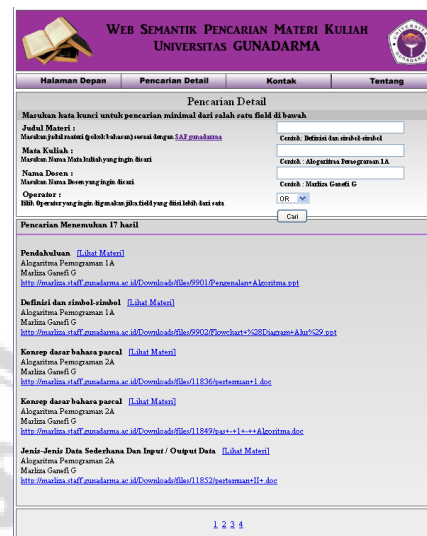
Uji coba selanjutnya pada pencarian detail dilakukan dengan cara hanya memasukan 1 kata kunci berupa nama mata kuliah pada pencarian detail yaitu fisika dan kimia. Berikut tampilan ketika pengujian pada pencarian detail.



Gambar 14. Hasil Pencarian Detail dengan kata kunci fisika dan kimia

Setelah dilakukan pencarian dengan menggunakan kata kunci “fisika dan kimia” pada mata kuliah, pada gambar diatas menunjukkan bahwa terdapat 48 hasil yang berhubungan dengan mata kuliah fisika dan kimia.

Uji coba yang terakhir pada pencarian detail dilakukan dengan cara memasukan kata kunci berupa nama dosen yang mengajar salah satu mata kuliah. Sebagai contoh pada pencarian ini menggunakan nama dosen “Marliza”. Berikut merupakan hasil pencarian menggunakan pencarian detail.



Gambar 15. Hasil Pencarian Detail dengan kata kunci Marliza

Pada gambar diatas menunjukkan hasil dari pencarian yang dilakukan menggunakan kata kunci berupa “marliza” pada field dosen. Pencarian tersebut menghasilkan 17 hasil yang berhubungan dengan marliza sebagai dosen yang mengajar suatu mata kuliah.

## Hasil Uji Coba Dari Sisi Pengguna

Untuk mengetahui penilaian dan tanggapan Pengguna terhadap aplikasi yang dikembangkan, dilakukan ujicoba terbatas dan penyebaran kuesioner penilaian. Dalam ujicoba terbatas dan penyebaran kuesioner kepada mahasiswa gunadarma sejumlah 20 orang. Kuesioner yang digunakan menggunakan Aspek-aspek penilaian yang berasal dari teori *evaluation web* dengan menggunakan konsep *seven checklist*. Konsep tersebut digunakan untuk melakukan evaluasi terhadap suatu web dengan tujuh komponen penilaian. Pada kuisisioner yang digunakan untuk mengevaluasi aplikasi ini hanya mengambil

empat dari tujuh aspek yang tersedia. Aspek yang dinilai antara lain tampilan umum, kemudahan penggunaan, kegunaan serta validasi link.

Tabel 2. Hasil Uji Coba Pengguna

N o	Aspek Penilai an	Jum lah Soal	Jumla h Respo nden	Perol ehan Skor	Perse ntase
1	Tampil an Umum	3	20	223	74,3%
2	Kemud ahan Penggu naan	3	20	195	65%
3	Kegun aan	3	20	244	81,3%
4	Valida si link	1	20	82	82%

Pada tabel hasil uji coba diatas terlihat bagaimana respon pengguna terhadap aplikasi ini. Pada aspek-aspek penilaian diatas terlihat bahwa aspek kegunaan dan validasi link mendapatkan penilaian yang cukup baik. Sedangkan pada aspek kemudahan penggunaan mendapat penilaian yang terkecil, dikarenakan kurangnya beberapa fitur pendukung untuk mempermudah pencarian.

### Perbandingan dengan Metode lain

Penggunaan metode penelitian pada aplikasi ini mempunyai beberapa kekurangan dan kelebihanya dibandingkan dengan menggunakan metode lain. Pada metode ini menggunakan metode semantik web dimana representasi data yang tersedia dapat diubah

kedalam format yang dapat dimengerti oleh mesin. Sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan informasi dengan lebih baik. Selain itu penggunaan RAP sebagai framework yang digunakan untuk mengolah ontologi yang tersedia kedalam PHP atau dalam bentuk HTML mempermudah pembuatnya mengatur atau memanipulasi data yang diperlukan agar sesuai dengan kebutuhan informasi yang dibutuhkan. Namun pada metode ini memiliki beberapa kekurangan antara lain, pencarian hanya dapat dilakukan berdasarkan kata kunci yang tersedia. Jika pencarian dilakukan diluar knowledge yang tersedia maka tidak akan menghasilkan keluaran yang diinginkan. Kekurangan yang kedua adalah penggunaan RDF sebagai ontologi masih menggambarkan data yang tersedia secara sederhana dan belum dapat memperlihatkan relasi yang cukup detil dari data-data yang tersedia.

## PENUTUP

### Kesimpulan

Aplikasi pencarian materi perkuliahan ini dapat melakukan pencarian materi perkuliahan dengan menggunakan kata kunci berupa judul materi, mata kuliah serta nama dosen yang mengajar materi tersebut. Pencarian juga dapat dilakukan dengan menggunakan satu kata kunci atau beberapa kata kunci. Penggunaan teknologi semantik web pada aplikasi ini bertujuan agar representasi data yang tersedia dapat diubah kedalam format yang dapat dimengerti oleh mesin. Sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan informasi dengan lebih baik.

Pemilihan RAP sebagai penghubung antara ontologi RDF yang ada dengan tampilan yang ada membuat manipulasi keluaran untuk user lebih mudah dilakukan sehingga dapat menghasilkan keluaran yang mudah dimengerti oleh pengguna.

Pada aplikasi yang sudah di uji coba terlihat bahwa tampilan dan kegunaan aplikasi tersebut dinilai cukup baik oleh pengguna . Namun pada faktor kegunaan masih dirasa kurang, karena belum tersedianya beberapa fitur yang mendukung kemudahan dalam melakukan pencarian.

### Saran

Dalam pembuatan aplikasi pencarian aplikasi materi perkuliahan menggunakan semantik web dengan ontologi RDF dan framework RAP masih memiliki beberapa kekurangan antara lain terlihat pada hasil uji coba yang dilakukan oleh pengguna pada bagian kemudahan penggunaan mendapatkan hasil yang paling terkecil. Hal tersebut disebabkan belum fasilitas-fasilitas yang membantu penggunaannya. Seperti fasilitas daftar materi perkuliahan yang tersedia serta fasilitas *auto complete* pada *field* yang digunakan untuk kata kunci pencarian. Selain itu masih kurangnya representasi data yang tersedia sebagai ontologi untuk pencarian materi perkuliahan.

### DAFTAR PUSTAKA

[1] Siahaan, Daniel O,. 2006. *Graphical Notations For Semantic Web Language*. Institut Teknologi Sepuluh Nopember, Surabaya.

[2] Muslimin , Amiril., Waskitho Wibisono, dan Daniel O Siahaan. 2006.

*Image Search Engine Using Semantic Web*. Institut Teknologi Sepuluh Nopember. Surabaya.

[3] Marja, Riitta Koivunen & Miller, Eric. Agustus 2011. *W3C Semantic Web Activity*. <http://www.w3.org/2001/12/semweb-fin/w3csw>.

[4] Dumbill , Edd. Agustus 2011. *The Semantic Web: A Primer*. <http://www.xml.com/pub/a/2000/11/01/semanticweb/index.html>.

[5] Pramudiono, Iko. Agustus 2011. *Semantik Web : GenerasiBbaru WWW*, <http://www.beritaipetek.com/zberita-beritaipetek-2006-05-15-Semantic-Web:-Generasi-Baru-WWW.shtml>.

[6] Wicaksana, I Wayan Simri. 2006. *Ontology: Bahasa dan Tools Protégé*. Universitas Gunadarma. Depok.

[7] Newman, David R. 2010. *The Building and Application of a Semantic e-Research Society*. University of Southampton.England.

[8] Anonim. Agustus 2011.*RDF Tutorial*. <http://www.w3schools.com/rdf/default.asp>.

[9] W3C Recommendation. Agustus 2011. *RDF:Concept and Abstract Syntax*. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.2004>.

[10] Awaludin , Muhammad. 2009. Sistem Navigasi dan Pencarian Berbasis Konteks Pada Konten E-Learning Menggunakan Teknologi Web Semantik. Institut Teknologi Sepuluh November. Surabaya.

[11] Ali. September 2011. Kelebihan dan Kekurangan Google. <http://alibwzr.blogdetik.com/2009/03/09/kelebihan-dan-kekurangan-google/>.